

Complexity and Chaos Analysis of a Predator-Prey Ecosystem Simulation

Yasaman Majdabadi Farahani
 Department of Computer Science
 University of Windsor
 Windsor, Canada
 majdaba@uwindsor.ca

Abbas Golestani, Robin Gras
 Department of Computer Science
 University of Windsor
 Windsor, Canada
 golesta@uwindsor.ca, rgras@uwindsor.ca

Abstract— We investigated the complexity level of an agent-based predator/prey ecosystem simulation. The variations of the times series associated to this ecosystem simulation are the result of complex simulation mechanisms. For the purpose of understanding how close our system is to random or chaotic processes, we compare these data with data generated by a Markov chain as a simple process. The parameters of the corresponding Markov matrix are learned from the data generated by our simulation. Then we used the Markov chain to generate data similar to those of the simulation. We show that the Markov chain for all three orders, which we tested, generated prey and predator time series that are more random than their counterpart in the original simulation. Also, we used the largest Lyapunov exponent to determine the chaotic behavior of the simulation. We discuss the largest Lyapunov exponent values for population time series of both prey and predator agents, which indicates chaotic behavior in our agent-based ecosystem simulation.

Keywords- agent-based ecosystem; chaos analysis; complexity analysis; Markov chain

I. INTRODUCTION

Few attempts have been made to model a complete ecosystem and analyze its complex behavior using an agent-based approach. A predator-prey model proposed by Ward et al. [4] in which the agent model is dedicated to represent schooling behaviors, and the evolution is an offline mechanism using a genetic algorithm. More recently, Ronkko [5] has proposed a high-scale simulation based on a particle system approach. There is, however, no evolution mechanism in this artificial ecosystem. More works has been done on the Avida platform [2], which proposes self-replicating and evolving digital organisms. Each digital organism consists of a virtual CPU that processes a sequential program. The biological complexity in these organisms has been defined by Huang et al [3] as the generic information, which an organism has about its environment. However, none of these papers discuss about the complexity of the overall behavior of the simulation.

We are interested in analysing the complexity of such complex dynamic system. We have created a generic platform capable of simulating complex ecosystem with intelligent agents interacting and evolving in a large and dynamic environment [1]. This is the only simulation modeling the fact that agent behaviors affect evolution and speciation. The agents display very complex behavior using

Fuzzy Cognitive Map (FCM) model [19] to make decisions. We would like to understand how predictable the complex system we have conceived is. The two opposite and extreme situations that lead to a rather unpredictable system are random processes and chaotic processes. Therefore, we would like to investigate how close our complex system is to these two extremes.

The rest of the article is organized as follows. We first review our ecosystem simulation, which uses FCM as a behavior model in Section 2. Predicting population using the Markov chain is explained in Section 3. The Markov chain, transition matrix, and predicting prey and predator population is described in this section as well. The Lyapunov exponent is described in Section 4, and experiments and results are shown in Section 5. Finally, in Section 6, we draw conclusions about this work and propose an extension to it.

II. INDIVIDUAL-BASED EVOLVING PREDATOR-PREY ECOSYSTEM SIMULATION USING FUZZY COGNITIVE MAPS AS A BEHAVIOR MODEL

In this section, the main parts of the already existing predator/prey ecosystem is briefly introduced.

A. Fuzzy Cognitive Maps

In general, fuzzy cognitive maps (FCMs) aim to represent the causal relationship between concepts, and to analyze inference patterns (the final states of the system after convergence). Formally, an FCM is a graph which contains a set of nodes C , each node C_i being a concept, and a set of edges I , each edge I_{ij} representing the influence of the concept C_i on the concept C_j . A positive weight associated with the edge I_{ij} corresponds to an excitation of the concept C_j from the concept C_i , whereas a negative weight is related to an inhibition (a zero value indicates that there is no influence of C_i on C_j). An activation level a_i is associated to each concept. An FCM allows computing the new activation levels of the concepts of an agent, based on its perception and on the current activation levels of its concepts.

B. Agents and Behavior Model

The agents of this simulation are either prey or predators, which act in a dynamic environment with 1000×1000 cells. Each cell may contain several individuals and some amount of food. Each agent has several properties that determine its physical capabilities and its behaviors. The behaviors are

determined by the interaction between the FCM and the environment.

Each agent possesses its own FCM that represents its genome. We use an FCM to model an agent's behavior (structure of the graph) and to compute the next action of the agent (dynamic of the map). In each FCM, three kinds of concepts are defined: sensitive (such as distance to foe or food, amount of energy, etc), internal (fear, hunger, curiosity, satisfaction, etc) and motor (evasion, socialization, exploration, breeding, etc.). The activation level of a sensitive concept is computed by performing a fuzzification of the information the agent perceives in the environment. For an internal concept, the activation level corresponds to the intensity of an internal state of the agent. Note that it enables to distinguish between perception and sensation: the sensation is the real value coming from the environment, and the perception is the sensation modified by the internal states. Activation levels of the motor concepts are used to determine what the next action of the individual will be. The amplitude of the chosen action is then calculated by performing a defuzzification of the value of the corresponding motor concept.

The FCM of an agent is transmitted to its offspring after being combined with the one of the other parents, and after the possible addition of some mutations. The behavior model of each agent is therefore unique.

C. Update

The time step represents a relatively long period of time, during which agents perform several small actions, which are summarized by a unique high level action. The possible high level actions for the agents are:

1. Evasion (for prey only), which is in the opposite direction of the closest foe within the vision range of the prey. The new position of the prey is computed using the speed of the prey.
2. Search for food, which is near the closest food (grass or meat) within the vision range.
3. Socialization, which is the direction toward the closest possible mate within the vision range.
4. Exploration in which the agent moves at its speed in the random direction.
5. Resting in which nothing happens.
6. Eating, which includes the update of the grass/meat unit in the cell and agents energy and hunger level.
7. Breeding: If the energy levels of both agents are more than a certain threshold, and they both choose the breeding action, then breeding is done.

For each action which requires the agent movement, its speed is computed proportional to the current activation level of the motor concept associated with its action.

At each time step, the values of the states of all the parameters in the model are updated. The three successive phases of the update process are as follows for all agent: Perception of the environment, computation of all its concepts, application of their action and update the energy level. Then some general updates for the whole world are

performed such as updating the species and updating the amount of food available in each cell of the world.

An agent has a quite short lifespan (in terms of number of time steps), and performs only a few dozens of actions during its life. This enables us to obtain a high level of population renewal, which is an important criterion for studying an evolutionary process.

Fig. 1 shows the population of prey and predator agents after each time step. As expected with a predator-prey system, it is clear that there is a dependency between the number of prey and the number of predators. The evolution of the number of predators follows that of prey, and vice versa. As a clear period, consider time steps between 2000-4000. When the number of prey grows, the number of predators also grows a few time steps later. But when the number of predators grows too much, the number of preys decreases a few time steps later, leading to a still-later decrease in the number of predators.

D. Evolution

In our simulation, evolution stems from several mechanisms: mating, mutation and speciation. Since species membership of the agents is evaluated at each time step, births and deaths of individuals influence the general species composition. Thus, a species can emerge or disappear at any time step. This enables us to model the evolution of populations of individuals sharing important genetic properties. Due to our species model, species evolution is derived directly from individual evolution. If the mating is successful, the two parents give birth to a unique offspring. This offspring inherits a combination of the genomic information of its parents, with possible mutations and crossover. The genome of an agent is defined as the set of edges, associated with their weights, of its FCM. More precisely, for each concept, the child inherits all the incident edges of this concept from one of its two parents. During that process, the weights of the edges can be modified, based on a probability of mutation (which is a parameter of the simulation). Moreover, some new edges can be created; and some old edges can be removed (if their weight becomes smaller than a given threshold). The apparition of new edges is a very important mechanism, in the sense that new influences between concepts can emerge during the evolutionary process. This allows the apparition of more complex and potentially more adaptive behaviors.

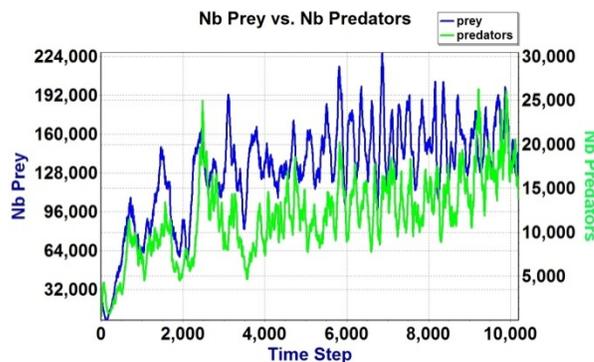


Figure 1. Population of prey and predator agents

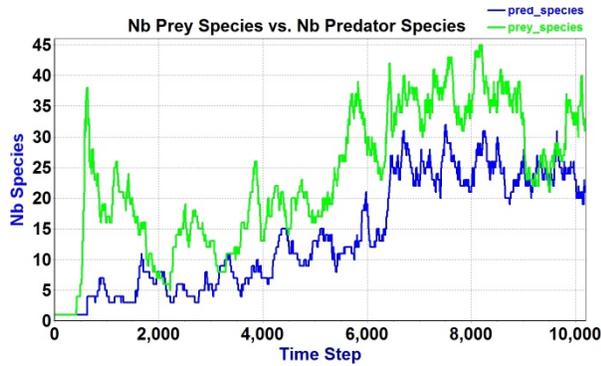


Figure 2. Evolution of the prey and predator species

In general, a single mutation is neutral and cannot produce a very different behavior model. It is the accumulation of neutral mutations during several time steps that allows the apparition of new individual behaviors, and then of new species.

Fig. 2 depicts the evolution of the prey and predator species in different time steps. Comparing Fig. 1 and Fig. 2, it appears that the number of species is closely correlated with the number of agents.

These are the kind of data that we would like to analyze. We consider several time series depicting the variation of different quantities in our system and define a protocol to evaluate the complexity level of our system by an analysis of these time series.

III. PREDICTING POPULATION USING THE MARKOV CHAIN

As it appears in Section 2, the agents in our simulation display a very complex behavior using FCM to make a decision. Moreover, the whole behavior of the ecosystem is a very complex system involving interaction between hundreds of thousands of complex agents. However, as it has been shown in [1], the overall system presents interesting correlation patterns. For instance, the population of prey or predators has strong correlations with the number of predators, prey, level of grass and meat, species distribution, and so on. Despite such regularities, the simulation is far from being easily predictable. The amplitudes and times of inflation and deflation vary considerably, but their correlation is conserved. This means we can suppose that there is no easy way, excluding the simulation itself, to predict the state of the system at time step t , knowing the state of the system at time step $t-1$. However, the variations of the time series associated to these numbers are not random as they are the results of the application of the simulation mechanisms. We are interested in evaluating how complex the time series generated by the simulation are. We would like to understand how predictable the complex system we have conceived is. The two opposite and extreme situations that lead to a system, which is hardly predictable are random processes and chaotic processes. Therefore, we would like to investigate how close to these two extremes our complex system is. In that purpose, we have compared the data generated by our system with data generated by a simpler process that generates data similar to the ones of the

simulation. We used a Markov chain as a simple model of our simulation. We have learned the parameters of the corresponding Markov matrix from the data generated by our simulation. Then we used the Markov chain to generate data similar to those of the simulation. We have then compared the time series data generated by both processes and measured their respective level of randomness/chaos.

A. Markov Chain

Define Suppose we generate a sequence of random variables, $\{X_0, X_1, X_2, \dots\}$, such that at each time t , the next state X_{t+1} depends only on the current state of the chain, X_t . This sequence is called a Markov chain, which is formally denoted as follows [7].

$$Pr(X_{t+1} = x_{t+1} | X_t = x_t, \dots, X_0 = x_0) = Pr(X_{t+1} = x_{t+1} | X_t = x_t)$$

Now suppose we generate a sequence of random variables, $\{X_0, X_1, X_2, \dots\}$, such that at each time t , the next state X_{t+1} depends on the current state of the chain, X_t and the previous state of the chain, X_{t-1} . This sequence is called a 2nd order Markov chain [7].

$$Pr(X_{t+1} = x_{t+1} | X_t = x_t, \dots, X_0 = x_0) = Pr(X_{t+1} = x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1})$$

Similarly the 3rd order Markov chain is a sequence satisfying

$$Pr(X_{t+1} = x_{t+1} | X_t = x_t, \dots, X_0 = x_0) = Pr(X_{t+1} = x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2})$$

B. Transition Matrix

Consider a Markov chain with finite state space $\{1, 2, \dots, k\}$. A transition matrix describes the probability of moving from state i to j at each time step using $Pr(j|i) = P_{i,j}$; in other words, the i th row and j th column element of the transition matrix P is given by $Pr(j|i)$ [6].

$$P = \begin{bmatrix} P_{1,1} & \dots & P_{1,k} \\ \vdots & \ddots & \vdots \\ P_{k,1} & \dots & P_{k,k} \end{bmatrix}$$

where the probabilities of each row sum up to 1.

$$\sum_j P_{i,j} = 1$$

because the overall probability of transitioning from state i to one of all possible states must be 1.

C. Markov Chain for Prey/Predator

To show the non-random behavior of the simulation, we have generated the transition matrix for the population of

prey and predator, and used the matrix probabilities to obtain the artificial population. The overall algorithm includes the three following steps:

- 1) Pre-processing
- 2) Transition matrix computation
- 3) Running the Markov chain

The first step includes the smoothing and quantization of the data set. The first part, smoothing, is performed on the dataset by the following linear transformation, which smoothes the original data while preserving its general characteristics and trend.

$$X_t = \frac{10x_t + 5x_{t+1} + 5x_{t-1} + 2x_{t+2} + 2x_{t-2} + x_{t+3} + x_{t-3}}{26}$$

where x_t is prey/predator population size at time t .

To build the transition matrix for the Markov chain, we need to have a finite state space, which means the number of the quantities should be constant and discrete. The interval for the quantization is computed by:

$$Quant_Interval = floor\left(\frac{(max(X) - min(X))}{k}\right)$$

where $max(X)$ and $min(X)$ are the maximum and minimum values seen in the prey or predator datasets, correspondingly, and k is the number of quantities, which is equal to the number of transition matrix rows in the 1st order Markov chain. The value of k is selected small enough to ensure the nonzero value for quantization interval. The smoothed values are transformed to the quantized data set by the following formula to have a minimum value of 1.

$$X'_t = floor\left(\frac{(X_t - min(X))}{Quant_Interval}\right) + 1$$

where X'_t is the quantized value of the smoothed value at time t .

Transition matrix computation depends on the order of the Markov chain. The transition matrix dimension is equal to the order of the Markov chain plus 1. In the 1st order Markov chain, the transition matrix is 2D, which includes the population at the current time and the interested population at the next time. In the 2nd order Markov chain, the matrix has another dimension, which is the previous population of prey or predator.

Each probability for this matrix is computed by counting how many times the same consecutive population sizes happening in the quantized data set.

By running the Markov chain from an initial value (or several previous values for higher order Markov chain), the artificial data set is created; the next population for each time instance is computed based on the current and previous populations (for the higher order of the Markov chain), and the corresponding probabilities in the transition Matrix.

IV. LYAPUNOV EXPONENT

Nonlinear signal processing is an important research area with many applications. Specifications and identifications of nonlinear signals can help us to detect nonlinear behavior of dynamical systems [8]. One specification, the discrimination of stochastic and chaotic behaviors of nonlinear time series, is a basic topic in nonlinear dynamic fields [9]. This specification has attracted researchers for a long time [10,12]. As many scientists believe that the natural phenomena have to be considered as deterministic and chaotic systems, it is important that a simulation used to model such a phenomenon generate a complex chaotic pattern [11]. For this reason fractal dimensions and Lyapunov exponents are the most prominent candidates to characterize the chaotic behavior [13], because they express complexity and predictability of a process and are a measure for chaos [14, 15, 16]. In this paper the Lyapunov exponent has been used.

Most experts would agree that chaos is the aperiodic, long-term behavior of a bounded, deterministic system that demonstrates sensitive dependence on initial conditions. For that purpose, we must quantify the sensitivity [17].

Lyapunov exponents quantify the exponential divergence of initially close state-space trajectories and estimate the amount of chaos in a system [18]. A bounded dynamic system with a positive Lyapunov exponent is chaotic [17].

Imagine two nearby initial points X_0 and $X_0 + \Delta X_0$, respectively. After one iteration of the map, the points are separated by

$$\Delta X_1 = f(X_0 + \Delta X_0) - f(X_0) \cong \Delta X_0 \hat{f}(X_0)$$

where $\hat{f} = df/dX$. Now, we define the local Lyapunov exponent λ at X_0 such that $e^\lambda = |\Delta X_1/\Delta X_0|$, or

$$\lambda = \ln|\Delta X_1/\Delta X_0| = \ln|\hat{f}(X_0)|$$

To obtain the largest Lyapunov exponent, we average the above equation over large enough iterations.

$$\lambda = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} \ln|\hat{f}(X_n)|$$

The largest Lyapunov exponent determines the average exponential rate of separation of two nearby initial conditions, or the average expansion of the space. A positive value shows chaos [17].

The different methods that have been proposed for computing Lyapunov exponents from time series can be divided into two classes: Jacobian-based methods and direct methods.

Direct methods directly estimate the divergent motion of the reconstructed states without fitting a model to the data [20]. The method, which has been used in this paper was proposed by Sato et al. [21], and Kurths and Herzel [22]. The average exponential growth of the distance of neighboring orbits is studied on a logarithmic scale, this time via the prediction error below

$$p(k) = \frac{1}{Nt_s} \sum_{n=1}^N \log_2 \left(\frac{\|y^{n+k} - y^{nn+k}\|}{\|y^n - y^{nn}\|} \right)$$

where y^{nn} is the nearest neighbor of y^n . The dependence of the prediction error $p(k)$ on the number of time steps k may be divided into three phases [19]. Phase 1 is the transient where the neighboring orbit converges to the direction corresponding to the largest Lyapunov exponent. During phase 2 the distance grows exponentially until it exceeds the range of validity of the linear approximation of the flow. Then phase 3 begins where the distance increases more slowly than exponentially until it decreases again because of folding in the state space. In phase 2, a linear segment with slope λ_1 appears in the $p(k)$ vs. k diagram. This allows an estimation of the largest Lyapunov exponent λ_1 [22]. Fig. 3 gives an example to determine the largest Lyapunov exponent λ_1 of data by this method [19].

V. EXPERIMENTS AND RESULTS

The simulation is implemented in C++ and has been run using the Narwhal cluster on the Sharcnet system and produced 32500 time steps. The resulting prey/predator population has been used as the input data for the Markov analysis explained in Section 3. The Markov chain analysis is implemented in Matlab 7.1 and is run on the AMD dual core processor 3.00 GHz with 3.00 GB RAM.

The analysis is performed on predator/prey's population starting from time step 10,000. This time was provided to ascertain that the simulation reaches its stabilization. The smoothing and quantization is performed on each of the prey and predator datasets, and 40 smoothed and quantized values are obtained. In Fig. 4 the time series analysis for prey dynamics is shown. This figure demonstrates the changes in the size of population in 10 time steps, in which x-axis represents the population value at time $t-10$, and y-axis represents the corresponding change in the size of the population at time t . Values between 1 and 40 in prey population corresponds to the smoothed and quantized values explained in Section 3.C. Specifically values 1 and 40 are the minimum and maximum values in this dataset, which are 44521 and 219007 respectively. Also each unit in y axis corresponds to the quantization interval given in the formula of Section 3.3, which is 4474 having k as 39.

As a simple example, consider the minimum value, 1, in prey population at time $t-10$, the changes in the prey population at the time t (10 time steps after), according to the Fig. 4 is 0 and 1 unites. In other words, if the value of smoothed and quantized prey population was 1, then the value of smoothed and quantized prey population 10 time steps after that would either remain unchanged or increase to 1 unit. As it can be seen form Fig. 4, prey dynamics has many variations around 0, meaning that in many time steps prey population size has remained the same in the next 10 time steps.

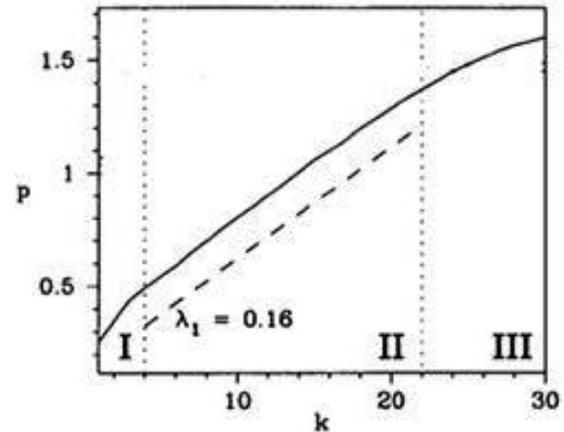


Figure 3. Prediction error p for experimental data vs. the number of time steps k . the slope of the solid line in the intermediate range of k gives the largest Lyapunov exponent $\lambda_1 = 0.16$.

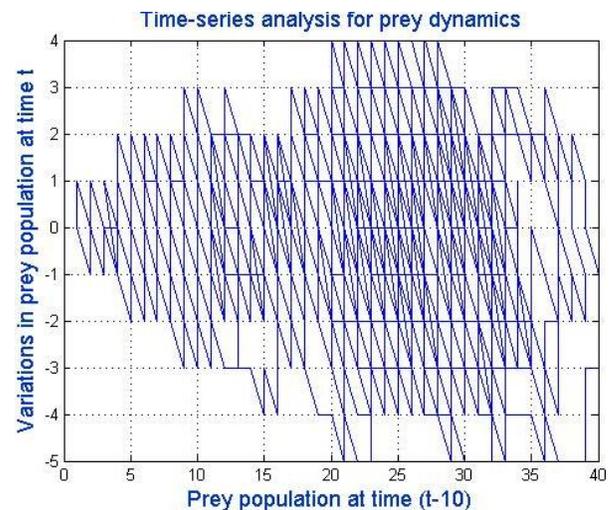


Figure 4. Time series analysis for prey dynamics

The lines in the graph represent transition between successive population states encountered during the simulation process.

Based on the values after preprocessing stage, frequency matrices and transition matrices are obtained. Markov runs are performed using these transition matrices to estimate the prey and predator populations starting with the first 10 values of the population of prey or predator in the simulation. The rest of the population values are estimated by applying the Markov run with the transition matrix. The Markov run for prey dynamics is shown in Fig. 5.

By comparing Fig. 4, which is the prey population dynamics in the simulation, and Fig. 5, which is the estimated prey population given the first 10 values of the simulation prey population, two important conclusions can be made. As expected the pairs of coordinates in Fig. 5 are the subset of pairs of coordinates in Fig. 4 meaning that some of the population pairs (population at time $t-10$ and its change at time t) are not generated during this Markov run.

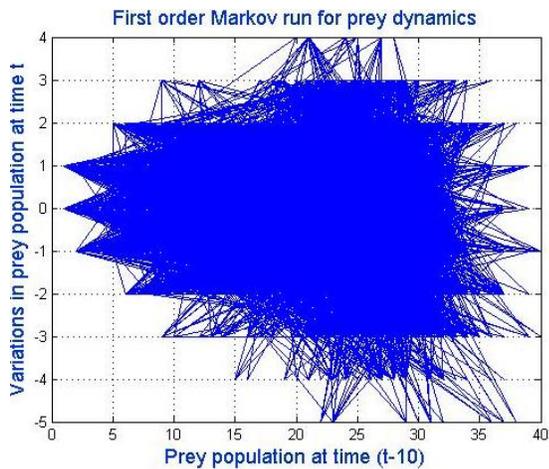


Figure 5. Markov run for prey dynamics

For instance, in Fig. 5, unlike Fig. 4, the pair (5, -2) did not appear meaning that if the population size at time $t-10$ was 5, the population size at time t would never be 3.

On the other hand, although the estimated values depicted in Fig. 5 are obtained using the transition matrix, which is based on the probabilities of values depicted in Fig. 4, the sequence of estimated values in Fig. 5 are clearly more random than the corresponding sequence represented in Fig. 4.

Higher order Markov chains are also implemented and the results are shown in Fig. 6 and Fig. 7.

Comparing Fig. 5 to 7 shows that the Markov chain for all of the order we have tested, generated prey dynamics that are more random than the original simulation prey dynamics. Note that although the average probabilities of having a certain number in the population size at the next interval increase by the order of the Markov chain, the average frequencies and zero rows percentage of the transition matrices are decreased (see Table 1). This is why the difference between different orders of Markov chain runs in population dynamics are not significant. A row in transition matrix with all zero values, indicate an undefined value for the next population interval. This row is created only in the 2nd or higher order Markov chains because it corresponds to the population sequence (sequence of 2 and 3 successive population values for 2nd and 3rd order Markov chain respectively), which was never appeared in the original population time series.

The same processes have been performed on predator time series and similar results have been obtained (Table 1).

To show the chaotic behavior observed in the population of prey and predator time series, the largest Lyapunov exponent values after different modifications of dataset are computed. In Table 2, the values of the largest Lyapunov exponent over simulation's prey and predator population data, smoothed time series and first, second and third order of the Markov run for the prey/predator population time series are presented.

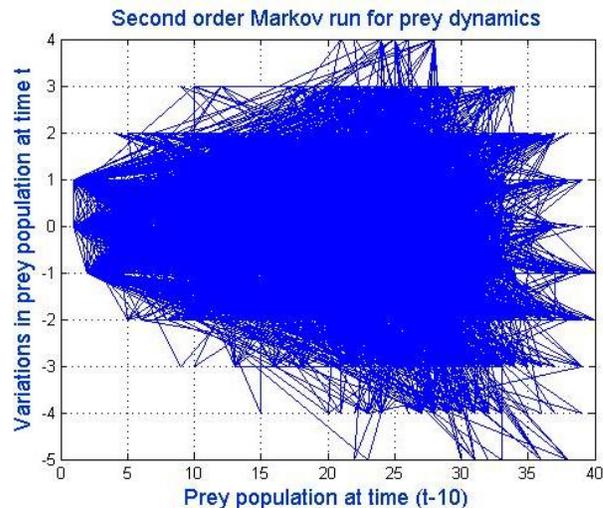


Figure 6. Second order Markov run for prey dynamics

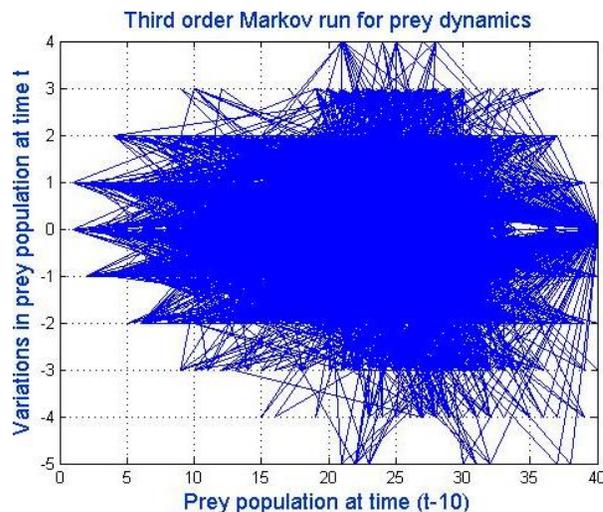


Figure 7. Third order Markov run for prey dynamics

The same processes have been performed on predator time series and similar results have been obtained (Table 1).

To show the chaotic behavior observed in the population of prey and predator time series, the largest Lyapunov exponent values after different modifications of dataset are computed. In Table 2, the values of the largest Lyapunov exponent over simulation's prey and predator population data, smoothed time series and first, second and third order of the Markov run for the prey/predator population time series are presented.

According to Table 2, we can conclude that the population time series, which has been produced by the simulation, indicate chaotic behavior because the largest Lyapunov exponent values for population time series are greater than zero. Obviously after the smoothing process, the largest Lyapunov exponent value is higher because smoothing removes random behavior. As we expected, the largest Lyapunov exponent of first, second and third order

TABLE I. DIFFERENCE BETWEEN MARKOV CHAIN ORDER VALUES IN TRANSITION MATRIX

		Zero Rows of Matrix	Average Probability of sequence	Average Frequency of sequence
Prey	1st-order Markov chain	0	0.17	101.67
	2nd-order Markov chain	0.14	0.27	27.83
	3rd-order Markov chain	0.013	0.37	10.18
Predator	1st-order Markov chain	0	0.14	79.54
	2nd-order Markov chain	0.18	0.22	17.28
	3rd-order Markov chain	0.021	0.34	5.9

TABLE II. LARGEST LYAPUNOV EXPONENT VALUES FOR DIFFERENT TIME SERIES

Data (time series)	Largest Lyapunov Exponent for prey	Largest Lyapunov Exponent for predator
Real data	0.5208	0.4862
Smoothed data	0.5603	0.5132
1st-order Markov chain	0.0117	0
2nd-order Markov chain	0	0
3rd-order Markov chain	0	0

Markov run for population time series is almost zero due to the random behavior of the Markov chain.

VI. CONCLUSION AND FUTURE WORK

We have conceived a protocol to evaluate the complexity level of our agent-based ecosystem simulation. The variations of the time series associated with the ecosystem simulation are the results of complex simulation mechanisms. To understand how close our system is to the random or chaotic processes, we have compared the data generated by our system with data generated by a Markov chain as a simple process.

As explained in Section 3, the parameters of the corresponding Markov matrix have been learned from the data generated by our simulation. Then we used the Markov chain to generate data similar to those of the simulation. As shown in Section 5 the Markov chain for all of the order we have tested, generates prey and predator time series that are more random than their counterpart in the original simulation.

We also used the largest Lyapunov exponent to determine the chaotic behavior of the simulation. Our experiments show that the largest Lyapunov exponent values for population time series of both prey and predator are positive, indicating a chaotic behavior in our ecosystem simulation, which is a good indication of a high complexity level.

As it appears in Section 4, the most prominent candidates to characterize chaotic behavior in a system are Fractal dimensions and Lyapunov exponents [13]. As the next step,

Fractal dimensions can also be applied on the time series produced by our ecosystem simulation to determine the complexity and predictability of the system and measure the chaotic behavior. We would like also to analyze other time series, depicting variations of other quantities like quantities of food or number of species, to have a better understanding of the overall complexity of our system.

ACKNOWLEDGMENT

This work is supported by the NSERC grant ORGPIN 341854, the CRC grant 950-2-3617 and the CFI grant 203617 and is made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET:www.sharcnet.ca).

REFERENCES

- [1] R. Gras, D. Devaurs, A. Wozniak, and A. Aspinall, "An Individual-based Evolving Predator-Prey Ecosystem Simulation using Fuzzy Cognitive Map as Behavior Model", *Journal of Artificial Life*, Number 4, Volume 15(4), pp. 423-463, 2009.
- [2] C. Ofria and C. Wilke. Avida, "A Software Platform for Research in Computational Evolutionary Biology", *Journal of Artificial Life*, Volume 10, pp. 191-229, 2004.
- [3] W. Huang, C. Ofria, and E. Torng, "Measuring Biological Complexity in Digital Organisms", Ninth International Conference on Artificial Life, Boston MA, Sept 12-15, 315-321, 2004.
- [4] C. R. Ward, F. Gobet, and G. Kendall, "Evolving Collective Behavior in an Artificial Ecology", *Journal of Artificial Life*, Volume 7(2), pp. 191-209, 2001.
- [5] M. Ronkko, "An artificial Ecosystem: Emergent Dynamics and Lifelike Properties", *Journal of Artificial Life*, Volume 13(2), pp. 159-187, 2007.
- [6] G. Latouche and V. Ramaswami, "Introduction to Matrix Analytic Methods in Stochastic Modelling", 1st edition, "Chapter 2: PH Distributions", Philadelphia, PA: ASA SIAM, 1999.
- [7] W. R. Gilks, Walter R. Gilks, Sylvia Richardson, and D. J. Spiegelhalter, "Markov Chain Monte Carlo in Practice", 1st edition, Chapman and Hall/CRC, 1996.
- [8] J. Hubbard and B. West, "Differential Equations: A Dynamical Systems Approach: Ordinary Differential Equations", Volume 5, Texts in Applied Mathematics. Springer, 1991.
- [9] Peter Grassberger, Thomas Schreiber, and Carsten Schaffrath, "Nonlinear Time Sequence Analysis", *International Journal of Bifurcation and Chaos*, Volume 1(3), pp 521-547, 1991.
- [10] Amir H. Omidvarnia and Ali Nasrabadi, "A New Irregularity Criterion for Discrimination of Stochastic and Deterministic Time Series", *Fractals*, Volume 16(2), pp. 1-12, 2008.
- [11] A. Golestani, M. R. Jahed Motlagh, K. Ahmadian, Amir H. Omidvarnia, and Nasser Mozayani, "A New Criterion for Distinguish Stochastic and Deterministic Time Series with the Poincaré Section and Fractal Dimension", *Journal of Chaos: An Interdisciplinary journal of Nonlinear Science*, CHAOS 19, Volume 19(1), pp. 1-13, March 2009.
- [12] L. Romanelli, M. A. Figliola, and F. A. Hirsch, Deterministic Chaos and Natural Phenomena. *Journal of Statistical Physics*, Vol. 53, Nos. 3/4, 1988.
- [13] J. Holzfuss and U. Parlitz, "Lyapunov Exponents from Time Series", *Lecture Notes in Mathematics*, Springer, Berlin, 1990.
- [14] J.P. Eckmann and D. Ruelle, "Ergodic Theory of Chaos and Strange Attractors", *Reviews of Modern Physics*, Volume 57(3), pp. 617-656, 1985.
- [15] W. Lauterborn and J. Holzfuss, "Evidence for a Low-Dimensional Strange Attractor in Acoustic Turbulence", *Physics Letters A*, Volume 115(8), pp. 369-372, 1986.

- [16] W. Lauterborn and J. Holzfuss, "Acoustic Chaos", International Journal of Bifurcation and Chaos, Volume 1(1), pp. 13-26, 1991.
- [17] Julien Clinton Sprott, "Chaos and Time Series Analysis", Oxford University Press, 2003.
- [18] Michael T. Rosenstein, James J. Collins, and Carlo J. De Luca, "A Practical Method for Calculating Largest Lyapunov Exponents from Small Data Sets", Journal of Physica D: Nonlinear Phenomena, Volume(65), pp. 117-134, 1993.
- [19] B. Kosko, "Fuzzy cognitive maps", International Journal of Man-Machine Studies, 24, 65-75, 1986.
- [20] U. Parlitz, "Nonlinear Time-Series Analysis", Nonlinear Modeling - Advanced Black-Box Techniques Eds. J.A.K. Suykens and J. Vandewalle Kluwer Academic Publishers, pp. 209-239, 1998.
- [21] Sato, S., M.Sano ,and Y. Sawada, "Practical methods of measuring the generalized dimension and largest Lyapunov exponent in high dimensional chaotic systems", Progress of Theoretical Physics, 77, pp. 1-5, 1987.
- [22] Kurths, J. and H. Herzel, "An attractor in solar time series", Physica D, 25, pp. 165-172, 1987.